DATA PROCESSING APPARATUS

BACKGROUND OF THE INVENTION

The present invention relates to a data processing

5   apparatus for executing the arithmetic processing of data in

which the bit width of one word is not standard 2^n bits.

Some of apparatuses for outputting or processing images,

voices etc. employ data in which the bit width of one word is

not standard 2^n bits, in order to enhance the qualities of the

10   images and the voices or to affix additional information to the

images and voices. Such apparatuses include, for example, a

so-called "third-generation portable telephone", and an

information processing apparatus which generates image data

having a plurality of gradations. A digital signal processor

15   (hereinbelow, termed "DSP") or any other data processing

apparatus is installed in such an apparatus, whereby various

items of arithmetic processing are executed by converting the

bit width of one word into standard 2^n bits.

Now, a presented data processing apparatus will be

20   described by taking as an example the DSP which is installed

in the third-generation portable telephone.

Using the DSP, the third-generation portable telephone

extracts the signals of several specified bandwidths from within

a broad frequency bandwidth in order to favorably communicate

25   at all times. Besides, from among the specified bandwidths,

one of high reception sensitivity is especially selected for the communications. Incidentally, on this occasion, the DSP usually extracts the signals of the specified bandwidths by employing a technique called "digital matched filter 5 (hereinbelow, termed "DMF") algorithm", but the technique itself is not pertinent to the subject matter of the present invention and shall be omitted from detailed description here.

In the presented data processing apparatus, data of 16 bits are outputted from a memory to an ALU at one time. However, 10 data for use in arithmetic processing correspond only to 10 of the 16 bits. Therefore, the presented data processing apparatus wastefully outputs the data of 6 bits to the ALU at one time.

Moreover, the ALU has a built-in arithmetic unit of 32-bit width, but it uses only 10 bits in the 32-bit width. Therefore, 15 the presented data processing apparatus wastes the arithmetic unit in correspondence with a 22-bit width.

With the intention of solving the drawbacks, there has been proposed a technique wherein the width of a bus is doubled (to 32 bits) so as to output data of 32 bits from the memory 20 to the ALU, and the arithmetic unit is divided into higher-order 16 bits and lower-order 16 bits so as to arithmetically process the data in parallel. Even with this technique, however, the memory is used in correspondence with only 10 bits in spite of the 16-bit width thereof. Therefore, the presented data 25 processing apparatus wastes 6 bits in the use of the memory.

In this manner, in the case where the arithmetic processing is done using the data of which one word does not have the standard bit width, the presented data processing apparatus has the problem that, since a reserved part (free area) is arranged 5 between I-part data and R-part data, the built-in arithmetic unit of the ALU and the memory are wastefully used, so the processability of the arithmetic unit and the capacity of the memory cannot be effectively utilized.

10   SUMMARY OF THE INVENTION

The present invention may provide a data processing apparatus which can effectively utilize an arithmetic processability and a memory capacity.

A data processing apparatus according to the present 15 invention includes address generation circuit, an address translation circuit, a selection circuit, a memory, a shifter, an ALU and a mixing circuit. The address generation circuit generates a logical address for data including 10 bit imaginary part, 10 bit real part and 12 bit reserved part. The address 20 translation circuit translates the logical address into a physical address for data including 16 successive bits of the imaginary part and the real part. The selection circuit generates a selection signal in response to the logical address. The memory for storing 16 bits of data in accordance with the 25 physical address. The shifter shifts the received data from

the memory in response to the selection signal. The ALU processes an arithmetic operation in response to the shifter output. The mixing circuit mixes the data received from the memory and the ALU and outputs the mixed data to the memory.

5

BRIEF DESCRIPTION OF THE DRAWINGS

Fig. 1 is a diagram showing the construction of a data processing apparatus according to the present invention;

Fig. 2 is a diagram showing a logical address space;

10   Fig. 3 is a diagram showing a physical address space;

Fig. 4 is a table showing the relations of logical addresses to physical addresses;

Fig. 5 is a table showing shift magnitudes;

Fig. 6 is a diagram showing the construction of a mixing

15   circuit;

Fig. 7 is a diagram showing the shift operation of the mixing circuit in an example;

Fig. 8 is a diagram showing the shift operation of the mixing circuit in another example; and

20   Fig. 9 is a diagram showing the shift operation of the mixing circuit in still another example.

DETAILED DESCRIPTION OF THE INVENTION

Now, an embodiment of the present invention will be

25   described with reference to the drawings by taking as an example

a DSP which is installed in a third-generation portable telephone. Incidentally, the drawings illustrate the invention merely schematically to the extent that it can be understood. Besides, throughout the drawings, the same reference numerals are assigned

5   to common constituents, which shall not be described repeatedly.

Fig. 1 is a diagram showing the construction of a data processing apparatus according to the present invention, Fig. 2 is a diagram showing a logical address space, Fig. 3 is a diagram showing a physical address space, Fig. 4 is a table showing the

10   relations of logical addresses to physical addresses, Fig. 5 is a table showing shift magnitudes, Fig. 6 is a diagram showing the construction of a mixing circuit, and Figs. 7 - 9 are diagrams each showing the shift operation of the mixing circuit.

Incidentally, the data processing apparatus (namely, DSP)

15   according to this embodiment keeps even-numbered addresses stored in a first memory block 19 and odd-numbered addresses stored in a second memory block 21. Therefore, the DSP can simultaneously access the first memory block 19 and the second memory block 21, and a memory 17 can output data of 32 bits at

20   one time.

The DSP shown in Fig. 1 is such that an address translation circuit 31, a select-signal generation circuit 33 and a mixing circuit 35 are added to a presented construction.

In this embodiment, a data space to be handled by an ALU

25   13 (hereinbelow, termed "logical address space") and a data space

on the memory 17 (hereinbelow, termed "physical address space") are distinguished and processed. Herein, the address translation circuit 31 is disposed in order to facilitate translation from an address on the logical address space

5 (hereinbelow, termed "logical address") into an address on the physical address space (hereinbelow, termed "physical address").

Besides, regarding data read out by the ALU 13, only parts of data referred to must be rewritten. By way of example, in

10 a case where the ALU 13 has read out the data of the first line, a processed result generated by the ALU 13 is stored in the first line. On this occasion, I-part data I(1) and R-part data R(1) are required for the next arithmetic processing and therefore need to be temporarily saved. Therefore, the mixing circuit

15 35 which saves parts of the data referred to, and which merges them with the processed result of the ALU 13 so as to write back the merged result into the memory 17 is disposed in this embodiment.

Further, arithmetic processing should desirably be easily

20 executed by the ALU 13. In this embodiment, therefore, it is permitted to easily calculate shift magnitudes from logical addresses in order that the positions of data for use in the arithmetic processing may become constant.

The address translation circuit 31 generates an address

25 which actually uses in reading out data from the memory 17 or

writing data into the memory 17, in such a way that a value is added to an address outputted from a memory address generation circuit 9, the value being obtained by shifting the outputted address 2 bits. Incidentally, the address outputted from the

5 address generation circuit 9 is called "logical address" in this embodiment, and the generated address "physical address".

The select-signal generation circuit 33 generates a select signal which designates the adjustment of a phase by shifting data predetermined bits or which designates the changeover of

10 an internal circuit, and it supplies the select signal to a shifter 11 and the mixing circuit 35. Upon receiving the select signal, the shifter 11 adjusts the phase by shifting the data predetermined bits so that the head of the data may become the 0th bit. Also, upon receiving the select signal, the mixing

15 circuit 35 adjusts the phase so that the data may come to the original bit position (bit position in the case where the data was read out from the memory 17). By the way, in this embodiment, 2 bits on the LSB side of the data are employed as the select signal.

20 Logical addresses are so constructed that, as shown in Fig. 2, I-part data and R-part data each being of 10-bit width succeed, followed by a reserved part of 12-bit width. On the other hand, physical addresses are so constructed that, as shown in Fig. 3, I-part data and R-part data each being of 10-bit width

25 succeed, followed by I-part data and R-part data.

In the physical addresses, the reserved part of 12-bit width is deleted, and the I-part data and the R-part data each being of 10-bit width are inserted in an area where the reserved part existed. Therefore, the logical addresses and the physical

5   addresses are in the following relationship:

Each physical address has such a value that a value obtained by shifting the corresponding logical address 2 bits is added to this logical address.

Fig. 4 shows the offset magnitudes between logical

10 addresses and physical addresses. More specifically, it shows, for example, the relation between an address where the head of I-part data in the logical address space exists and an address where the head of I-part data in the physical address space as corresponds to the former address exists. The address where

15 the head of the I-part data in the physical address space exists, changes in accordance with the address where the head of the I-part data in the logical address space exists. By way of example, as shown in Fig. 4, addresses 0, 2, 4, 6, 8, 10, 12, 14, 16, 18 ··· where the heads of I-part data I0 - I9 in the logical

20 address space exist become addresses 0, 1, 2, 3, 5, 6, 7, 8, 10, 11 ··· in the physical address space. In terms of a calculation formula, the relations are expressed by the following equation (1):

$$A = (a/2) + [(a/2)/4] \quad \dots \ (1)$$

25 Here, letter A denotes a physical address, letter $\underline{a}$ denotes

a logical memory address in the case where data of 32-bit width

as consists of an imaginary part of 10-bit width, a real part

of 10-bit width and a reserved part of 12-bit width is stored

in a memory of 16-bit width, and symbol [x] denotes the Gaussian

5 symbol which signifies the maximum integer not exceeding $\underline{x}$.

Besides, Fig. 5 shows the relations between logical

addresses and shift magnitudes from the reference positions of

data referred to (in case of the example shown in Fig. 5, the

head of I-part data In at each logical address becomes the 0th

10 bit). More specifically, it shows, for example, the relation

of the shift magnitude between the bit position of the head of

I-part data in the logical address space and the bit position

of the head of I-part data in the physical address space. The

shift magnitude changes depending upon an address where the head

15 of the I-part data in the logical address space exists. By way

of example, as shown in Fig. 5, bit positions 0, 2, 4, 6, 8,

10, 12, 14, 16, 18 ⋯ of the heads of the I-part data I0 – I9

in the logical address space become addresses 0, 4, 8, 12, 0,

4, 8, 12, 0, 4 ⋯ in the physical address space. In terms of

20 a calculation formula, the relations are expressed by the

following equation (2):

$$B = (a/2) - 4[(a/2)/4] \quad \dots \quad (2)$$

Here, letter B denotes a bit position in the physical

address space, letter $\underline{a}$ denotes the logical memory address in

25 the case where the data of 32-bit width as consists of the imaginary

part of 10-bit width, the real part of 10-bit width and the reserved part of 12-bit width is stored in the memory of 16-bit width, and symbol [x] denotes the Gaussian symbol which signifies the maximum integer not exceeding x.

5   Incidentally, this relationship corresponds to the quadruple of 2 bits on the side of the least significant bit (hereinbelow, termed "LSB") of the logical address.

This relationship will be explained below as to a case, for example, the DSP reads out data consisting of I-part data

10   I6 and R-part data R6.

More specifically, the head of the data consisting of the I-part data I6 and R-part data R6 exists at a logical address 12 on the logical address space, but it exists at a physical address 7 on the physical address space. Therefore, the DSP

15   calculates on the basis of the relations of physical addresses to logical addresses as shown in Fig. 4, the fact that the head of the data consisting of the I-part data I6 and R-part data R6 exists at the physical address 7 on the physical address space, and it reads out the data consisting of the I-part data I6 and

20   R-part data R6, from the physical address 7 of the memory 17.

On this occasion, in order to facilitate arithmetic processing in the ALU 13, the DSP adjusts a phase in such a way that the data to be read out from the memory 17 is shifted predetermined bits on the basis of the relations of shift

25   magnitudes to logical addresses as shown in Fig. 5. By way of

example, the head of the I-part data I6 exists at the position of the 8th bit as reckoned from the 0th bit of the physical address 7, so that the DSP shifts the I-part data I6 rightwards to the amount of 8 bits by the shifter 11.

5   The address translation circuit 31 translates the address of data which is to be read out from the memory 17 or to be stored in the memory 17, in accordance with such relationships. Besides, the select-signal generation circuit 33 determines a position where a part for use in the arithmetic processing is extracted

10   from within the data read out from the memory 17, in accordance with such relationships.

The mixing circuit 35 mixes an arithmetically processed result outputted from the side of the ALU 13 along a first route 23, and data outputted from the side of the memory 17 along a

15   third route 23'.

The operation of the mixing circuit 35 will be explained below.

As shown in Fig. 6, the mixing circuit 35 is constructed like, so to speak, an assembly of registers and selectors.

20   More specifically, the mixing circuit 35 includes a first register group 41 which consists of three registers each having a 4-bit width and to which read data [11:0] (that is, data of 12 bits of bit positions 0 - 11) read out from the first memory block 19 is inputted, a second register group 43 which consists

25   of eight registers each having a 4-bit width and to which data

-11-

of 32-bit width outputted from the side of the ALU 13 (or an Acc 15) is inputted, and a third register group 45 which consists of three registers each having a 4-bit width and to which read data [15:4] (that is, data of 12 bits of bit positions 4 - 15)

5 read out from the second memory block 21 is inputted.

Besides, in the mixing circuit 35, the three registers of the first register group 41 and three registers on the side of the most significant bit (hereinbelow, termed "MSB") in the second register group 43 are connected through selectors.

10 Further, every four of the registers in the second register group 43, from the LSB side to the MSB side in this group, are connected through selectors. Still further, three registers on the LSB side in the second register group 43 and the three registers of the third register group 45 are connected through selectors.

15 Herein, each selector is connected with the select-signal generation circuit 33 so as to receive a select signal therefrom. In this embodiment, 2 bits on the LSB side of a logical address are employed as the select signal. When supplied with the 2 bits on the LSB side of the logical address from the select-signal

20 generation circuit 33, the selector changes-over a circuit in accordance with the content of the select signal.

The mixing circuit 35 operates as shown in Figs. 7 - 9 by way of example.

(Operating example 1: Case where DSP has accessed logical

25 address 0 (I0, R0))

-12-

In a case where the DSP has accessed a logical address 0 (I0, R0), the values of 2 bits on the LSB side of the logical address are "00". Therefore, a physical address becomes 0 in view of the relations of physical addresses to logical addresses

5   as shown in Fig. 4. On this occasion, as shown in Fig. 7, the mixing circuit 35 mixes outputs from the first - fifth registers on the MSB side of the second register group 43 and outputs from the three registers of the third register group 45.

By the way, in the operating example 1, the DSP operates

10   as stated below.

First, the DSP reads out I-part data I0 and R-part data R0 from the physical addresses 0 and 1 of the first and second memory blocks 19, 21. The shift magnitude of the shifter 11 on this occasion becomes 0 because the values of the 2 bits on

15   the LSB side of the logical address are "00". Therefore, the I-part data I0 and R-part data R0 which are read data are directly inputted to the ALU 13. On the other hand, also in the mixing circuit 35, the values of the 2 bits on the LSB side of the logical address become "00", and data (1) which are parts of the read

20   data and which are to be saved are generated. The data (1) which are generated on this occasion become $I1[9:0]$ and $R1[1:0]$. Since the data $I1[9:0]$ and $R1[1:0]$ will be used at later arithmetic processing, those $4th$ - $15th$ bits of the second memory block 21 at which the data $I1[9:0]$ and $R1[1:0]$ are stored must not

25   be overwritten at this stage.

Subsequently, the DSP executes arithmetic processing between the I-part data I0 as well as the R-part data R0 and another data, thereby to generate update data (2). Thereafter, the DSP outputs the update data (2) of the I-part data I0 and

5   R-part data R0 from the Acc 15. Subsequently, the mixing circuit 35 combines the data (1) and the update data (2), thereby to generate overwriting write data (3) for the first and second memory blocks 19, 21.

(Operating example 2: Case where DSP has accessed logical

10   address 1 (I1, R1))

In a case where the DSP has accessed a logical address 1 (I1, R1), the values of 2 bits on the LSB side of the logical address are "01". Therefore, a physical address becomes 1 in view of the relations of physical addresses to logical addresses

15   as shown in Fig. 4. On this occasion, as shown in Fig. 8, the mixing circuit 35 mixes an output from the first register on the MSB side of the first register group 41, outputs from the first - fifth registers on the MSB side of the second register group 43 and outputs from the first - second registers on the

20   LSB side of the third register group 45.

By the way, in the operating example 2, the DSP operates as stated below.

First, the DSP reads out I-part data I1 and R-part data R1 from the physical addresses 1 and 2 of the first and second

25   memory blocks 19, 21. The shift magnitude of the shifter 11

-14-

on this occasion becomes 4 because the values of the 2 bits on the LSB side of the logical address are "01". Therefore, values obtained by shifting the I-part data I1 and R-part data R1 which are read data, rightwards to the amounts of 4 bits, are inputted

5   to the ALU 13. On the other hand, also in the mixing circuit 35, the values of the 2 bits on the LSB side of the logical address become "01", and data (4) which are parts of the read data and which are to be saved are generated. The data (4) which are generated on this occasion become I2[7:0] and R2[9:0]. Since

10   the data I2[7:0] and R2[9:0] will be used at later arithmetic processing, those 8th - 15th bits of the first memory block 19 and those 0th - 11th bits of the second memory block 21 at which the data I2[7:0] and R2[9:0] are stored must not be overwritten at this stage.

15   Subsequently, the DSP executes arithmetic processing between the I-part data I1 as well as the R-part data R1 and another data, thereby to generate update data (5). Thereafter, the DSP outputs the update data (5) of the I-part data I1 and R-part data R1 from the Acc 15. Subsequently, the mixing circuit

20   35 combines the data (4) and the update data (5), thereby to generate overwriting write data (6) for the first and second memory blocks 19, 21.

              (Operating example 3: Case where DSP has accessed logical address 3 (I3, R3))

25   In a case where the DSP has accessed a logical address

3 (I3, R3), the values of 2 bits on the LSB side of the logical address are "11". Therefore, a physical address becomes 3 in view of the relations of physical addresses to logical addresses as shown in Fig. 4. On this occasion, as shown in Fig. 9, the mixing circuit 35 mixes outputs from the three registers of the first register group 41 and outputs from the first - fifth registers on the MSB side of the second register group 43.

By the way, in the operating example 3, the DSP operates as stated below.

First, the DSP reads out I-part data I3 and R-part data R3 from the physical addresses 3 and 4 of the first and second memory blocks 19, 21. The shift magnitude of the shifter 11 on this occasion becomes 12 because the values of the 2 bits on the LSB side of the logical address are "11". Therefore, values obtained by shifting the I-part data I3 and R-part data R3 which are read data, rightwards to the amounts of 12 bits, are inputted to the ALU 13. On the other hand, also in the mixing circuit 35, the values of the 2 bits on the LSB side of the logical address become "11", and data (7) which are parts of the read data and which are to be saved are generated. The data (7) which are generated on this occasion become I4[9:0] and R4[5:0]. Since the data I4[9:0] and R4[5:0] will be used at later arithmetic processing, those 0th - 15th bits of the second memory block 21 at which the data I4[9:0] and R4[5:0] are stored must not be overwritten at this stage.

Subsequently, the DSP executes arithmetic processing between the I-part data I3 as well as the R-part data R3 and another data, thereby to generate update data (8). Thereafter, the DSP outputs the update data (8) of the I-part data I3 and

5 R-part data R3 from the Acc 15. Subsequently, the mixing circuit 35 combines the data (7) and the update data (8), thereby to generate overwriting write data (9) for the first and second memory blocks 19, 21.

The mixing circuit 35 operates in this manner. Therefore,

10 the DSP need not perform unnecessary shift processing for the reason that, in case of the arithmetic processing, data to be arithmetically processed are stored in the same bit positions. It is accordingly possible to heighten the run speed of a program.

The present invention described above has the advantage

15 that data of 10-bit width can be formed without any free area even in a memory of 16-bit width, so the quantity of use of the memory can be enhanced. Besides, since the present invention is incarnated merely by adding the address translation circuit 31, the select-signal generation circuit 33 and the mixing

20 circuit 35 to the presented apparatus, a circuit scale is enlarged little.

Incidentally, the present invention is not restricted to the foregoing aspects of performance, but various applications and modifications are considered within a scope not departing

25 from the subject matter of the present invention.

-17-

The present invention thus far described has the advantage that data of 10-bit width can be formed without any free area even in a memory of 16-bit width, so the quantity of use of the memory can be enhanced.

5